Multi-processing as a model for high-throughput data acquisition (especially for airborne applications)

Robert Kremens, Juan Cockburn, Jason Faulring, Peter Hammond, Donald McKeown, David Morse, Harvey Rhody, Michael Richardson

Rochester Institute of Technology
Center for Imaging Science
Computer Engineering
ABSTRACT

High throughput data acquisition, whether from large image sensors or high-energy physics detectors, is characterized by a 'set-up' phase, where instrumental parameters are set, a 'wait' phase, in which the system is idle until a triggering event occurs, an 'acquisition' phase, where a large amount of data is transferred from transducers to memory or disk, and optionally, a 'readout' phase, where the acquired data is transferred to a central store. Generally, only the 'acquisition' phase is time critical and bandwidth intensive, the 'set-up' phase being performed infrequently and with only a few bytes of data transfer. A hardware trigger is often used because only hardware can provide the low degree of latency required for these high performance systems.

Previously, system designers have resorted to complex real time operating systems and custom hardware to increase system throughput. We propose a new multiprocessing acquisition model, where bandwidth from transducer to RAM or disk store is increased by the use of multiple general purpose computers using conventional processors, RAM and disk storage. Bandwidth may be increased indefinitely by increasing the number of these processing/acquisition/storage units.

We have demonstrated this concept using the Wildfire Airborne Sensor Program (WASP) camera system. Details of the design and performance, including peak and average throughput rates and overall architecture, will be discussed. Sustained acquisition throughputs of 20 MBytes/second have been obtained using a non-real time operating system (Windows XP) and IBM-PC compatible hardware in a three computer configuration. Plans for a generalized multi-processing acquisition architecture will be discussed.
Motivations for this work

- Funded by NASA to develop an airborne wildfire sensing camera using commercial, off the shelf technology (COTS)
- (BAD!) experience with previous airborne camera system using a single computer, custom Linux hardware drivers and custom Linux kernel
- Projected future need for high throughput data collection systems for terrestrial as well as airborne applications
Objectives

- Description of a typical airborne camera system
- Data bottlenecks in conventional architectures
- Describe our solution to the high bandwidth data recording problem
- Multi-processing architecture applied to these camera systems
- Description of the WASP airborne wildfire research camera
- Description of the MISI hyperspectral scanner system
- Show outstanding results!
What components comprise an airborne camera system?
Airborne data acquisition is by nature synchronous, event driven.

- System captures image frame as aircraft proceeds.
- Time interval between frames is constant and dependent on mission, speed of aircraft.
- Navigation information critical to reassembly of images into a mosaic.
Our airborne data systems have very high sustained data throughput rates

- WASP:
  - 3 X 655Kbytes images + 33.6MByte image in bursts every 2 – 4 seconds
  - Average throughput 9 – 18 MByte/sec
  - Maximum PCI bus speed 133Mbyte/sec (not sustained)
  - Typical mission – 100 ‘fields’ 355MByte total collection
  - Other peripheral data is also collected (navigation data)
- MISI
  - 96 X 180kHz X 2 bytes = 35 Mbyte/sec sustained without time gaps
  - Also collecting navigation data and doing some control
Single computer architectures have bus bandwidth (but not processing limitations) in our application

- **CPU**
- **Caches**
- **Front Side Bus (FSB)**
- **I/O Buses**
  - **NICs**
  - **Networks**
- **I/O Devices**
  - **Disks**
  - **Displays**
  - **Keyboards**

**Examples:**
- Alpha, AMD K7: EVE, 200-400 MHz
- Intel, P6, P6E: GTL+ 133 MHz
- Intel P4: 800 MHz

**I/O Buses:**
- PCI, 32-64 bits wide
- PCI-X, 64 bit

**Memory:**
- Off or On-chip

**Chipset:**
- North Bridge
- South Bridge

**I/O Subsystem**
We can increase throughput by increasing bus speed or width or using multiple computers

• New, wider, faster busses may be available
• Speed across bridges is under question
• Limited by speed of front side bus, at any rate
• Bus contention divides bus speed by ‘X’. This division ratio is application and operating system-dependent
• Hardware is probably not available on new, high speed busses
• Multiple computer architecture is very easy if systems do not require much inter-process communication
We can increase throughput by increasing bus speed or width or using multiple computers (2).

- We have a unique subset of data collection applications:
  - Our airborne data systems are really described as ‘multiple, independent processes using the same bus’
  - Very limited software communication between processes
  - Hardware triggering synchronizes various system elements
  - Begs for a multiple computer solution
In effect, we have hardware ‘objects’ with a common communication protocols

- Multiple camera instances with one controller
- Control computer synchronizes external events, provides user input and provides monitor and status functions
- Common control language (Ethernet messages) for:
  - Initialize
  - Arm (wait for trigger)
  - Report Trigger (and other status)
  - Readout
- Very general for all data acquisition operations
The ‘object oriented’ multi-processor data acquisition model
What about software?

- Multi-tasking operating systems not ideal for data collection
- Poor or no scheduling capability in most ‘conventional’ OSs
- Real time operating systems difficult to use - also hard to find competent programmers
- Hypothesis: given sufficient bus bandwidth, DMA, any OS should be adequate
- We chose Windows 2000/XP as an experiment
- It worked.
What about software? (2)

- We use Ethernet to connect all the computers and provide data and control communications
- Hard timing performed with hardware in the control computer
- Meta- and Ancillary- data collected in control computer
- File name synchronization to mesh data residing on different computer chassis
The multiprocessing architecture allows parallel software development and incremental hardware testing.

- Design goals:
  - Hardware/software modularity:
    - Each camera stands alone: ‘digital camera’ model
    - Parallel development of control system, and each camera
    - Camera systems expandable/replaceable
  - Low power consumption
    - 30 - 50 W camera
  - Commercial drivers for hardware
    - MS Windows
    - VB and C++ / Visual Studio
    - Flat-panel touch screen for I/O
  - Data rate consistent with < 0.2*PCI bus throughput
A little bit about the WASP camera

- Provides reliable day/night wildfire detection with low false alarm rate
- Provides useful fire detection map information in near real time
- Investigate new algorithms and detection methods using multispectral imaging

Phase 1
- Demonstrate sensor operation from an aircraft
- First flights very successful: semi-quantitative
- Geometric and radiometric calibration completed

Phase 2 (starting 1 October 2003)
- Automated on-board data processing including geo-referencing and fire detection
WASP uses many commercial components

COTS Camera for VNIR
- Proven aerial mapping camera
- 4k x 4k pixel format
- 12 bit quantization
- High quality Kodak CCD

COTS Cameras for SWIR, MWIR, LWIR
- Ruggedized industrial/aerospace equipment
- 640 x 512 pixel format
- 14 bit quantization
- < 0.05K NEdT

Measurement Accuracy
Position 5 m
Roll/Pitch 0.03 deg
Heading 0.10 deg
WASP uses 4 framing cameras in a scanning head.
Modular electronics allows expansion and maintenance

WASP Processor Architecture
Parallel - Modular
Robust - COTS

28V @ 12 A  250 lbs.

Control Computers
1 – Master
2 – High Res Vis
3 – IR Cameras

IR Camera Interface
Electronics (3)

Inertial Measurement System

Power Distribution

Batteries (if required for self-power)
WASP Fits nicely in our Piper Aztec
WASP II quick facts

- Resolution at 1500m AGL – 2m IR, 0.3m VIS
- Bands: visible (RGB or color IR) 0.9-1.8μm, 3-5μm, 8-11μm
- Frame rate: 0.5 Hz Vis, 30 Hz IR
- View Angle, 50° fixed, 110° scanning
- Coverage @ 120kts @3100m AGL: 100,000 ac/hr
- NEΔT: measured less than 0.05 K in MWIR
- Absolute geolocation accuracy~10m
- Geolocation repeatability: ~3m
WASP has detected a 15 cm charcoal test fire at 10,000 ft AGL.
WASP Performance and History

- WASP has the potential to serve a critical national need
  - reliable day/night detection and mapping of wildfires
- Successful operational season
  - 40+ flight days with no in-flight equipment failures
  - Verified detection of 6” charcoal fire at 10,000 ft AGL
  - Flew 6 calibration flights over charcoal beds and other targets
    - Test of geo- and ortho- rectification
    - Overload and gain setting
    - Resolution tests
  - Flew 3 prescribed fires in the Northeast
    - Vinton Furnace site at Wayne National Forest (Ohio)
    - Albany Pine Bush Preserve (TNC-NY)
- Internal camera model validated and image-to-image registration verified
- Flew 100,000 acre/hour mission (7GB) with 30 minute data delivery
- ‘Data pipeline’ processing started on our aircraft ‘super computer’
Typical WASP ortho- and geo- rectified data
The MISI hyperspectral imager is completely RIT designed

- 80 separate spectral bands
- UV to IR (0.3 to 14 microns)
- Scanner with multiple detectors
- 1800 pixels across track
- 1-3 m radian resolution
MISI makes even more severe bus bandwidth demands than WASP

- 96 - 12 bit data channels acquired simultaneously and continuously
- Complicated (and variable) trigger pulse generation: over 30 trigger signals required with 5 time events on each signal
- 4 Channels of RS232 for ancillary control functions
- Shutters, reference calibrators (3)
We solved the data throughput requirements by using multiple computers, massive memory and DMA transfer only.
Objectives

• Description of a typical airborne camera system
• Data bottlenecks in conventional architectures
• *Describe our solution to the high bandwidth data recording problem*
• Multi-processing architecture applied to these camera systems
• Description of the WASP airborne wildfire research camera
• Description of the MISI hyperspectral scanner system
• Show outstanding results!
Conclusions

- We have developed two high-throughput sophisticated airborne data acquisition units by pouring hardware on standard software
- We have shown the utility of multiple processor architecture and conventional OS/software for high performance computing
- We have certainly maximized price/performance by developing the system in ~4 man months of SW development time
- Need further experience in probing the limits of such architectures with regard to throughput and reliability
- We are now ready to apply the same techniques to other data collection problems